

COMPLETE SET OF PENDING CLAIMS

1-29. (Cancelled)

30. (Previously Presented) A data storage method comprising:
generating a virtual machine instruction sequence by compiler to be executed by a virtual machine;
dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block;
transmitting the instruction block to the virtual machine;
storing the instruction block in the virtual machine; and
formatting the instruction block to include:
an identifier area for storing an identifier that specifies a start position of the instruction block;
a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block;
a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block; and
each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas.

31-38. (Cancelled)

39. (Previously Presented) A data storage method comprising:
generating a virtual machine instruction sequence by compiler to be executed by a virtual machine;
dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block;
transmitting the instruction block to the virtual machine;
storing the instruction block in the virtual machine;
formatting the instruction block to include:
an identifier area for storing an identifier that specifies a start position of the instruction block;
a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block; and
a branch instruction area for storing at least one branch instructions belonging to the corresponding basic block.

40. (Previously Presented) A data storage method of Claim 39,
wherein the identifier of the instruction block is an address related information in the virtual machine instruction sequence.

41. (Previously Presented) A data storage method of Claim 40,
wherein the address related information is one of absolute address, relative address, and offset address.

42. (Previously Presented) A data storage method of Claim 40,
wherein whether each virtual machine instruction is positioned at a start position of the basic block is indicated by an address in the virtual machine instruction sequence to which the virtual machine instruction is allocated;
a virtual machine instruction at the start position of the basic block being allocated to a specific address in the virtual machine instruction sequence, and
a virtual machine instruction at other than the start position of the basic block being allocated to an address other than the specific address.

43. (Previously Presented) A data storage method of Claim 40,
wherein virtual machine instructions of the virtual machine instruction sequence each comprises:
an identifying unit for storing identification information which indicates if the virtual machine instruction is positioned at a start position of the basic block; and
an operation specifying unit for specifying an operation to be executed by the virtual machine.

44. (Previously Presented) A data storage method of Claim 40,
the basic blocks; and
identification tags, each designates an address related information of the virtual machine instruction at a start position of the basic block; attachment of the tag indicating if the virtual machine instruction corresponding to the identification tag is positioned at the start position of the basic block.

45-51. (Cancelled)

52. (Currently Amended) ~~A method comprising:~~ A machine-readable medium that provides instructions, which when executed by a processor, cause the processor to perform operations comprising:

generating a virtual machine instruction sequence by compiler to be executed by a virtual machine;

dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block;

transmitting the instruction block to the virtual machine;

a storing step of storing the instruction block in the virtual machine;

formatting the instruction block to include:

an identifier area for storing an identifier that specifies a start position of the instruction block;

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block;

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block; and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas,

wherein the division of the virtual machine instruction sequence into a plurality of separately identifiable instruction blocks having a single branch instruction area reduces the amount of branch destination processing that would otherwise be necessary with a single instruction sequence with branch instructions throughout.

53. (Currently Amended) ~~A method method comprising:~~ A machine-readable medium that provides instructions, which when executed by a processor, cause the processor to perform operations comprising:

generating a virtual machine instruction sequence by compiler to be executed by a virtual machine;

dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block;

transmitting the instruction block to the virtual machine;

storing the instruction block in the virtual machine;

formatting the instruction block to include:

an identifier area for storing an identifier that specifies a start position of the instruction block;

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block;

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block; and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas,

wherein the virtual machine instruction sequence is transmitted after being divided into the plurality of instruction blocks.